FIG.1

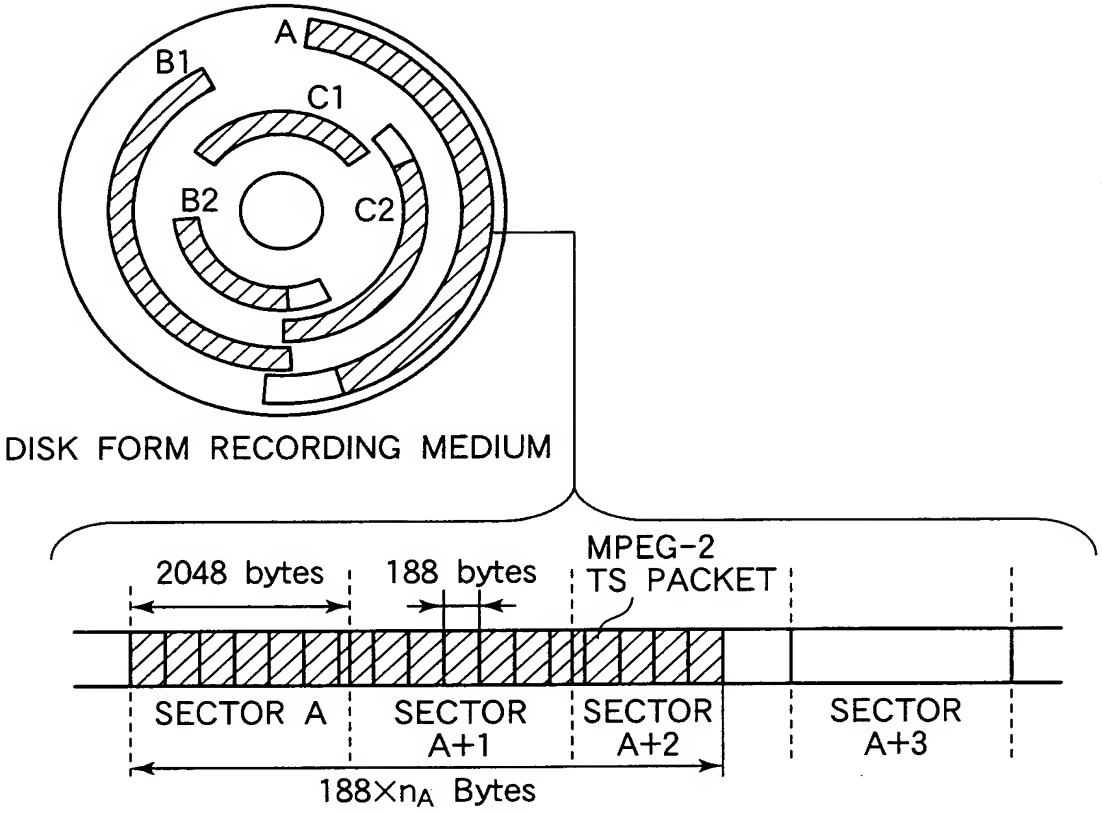
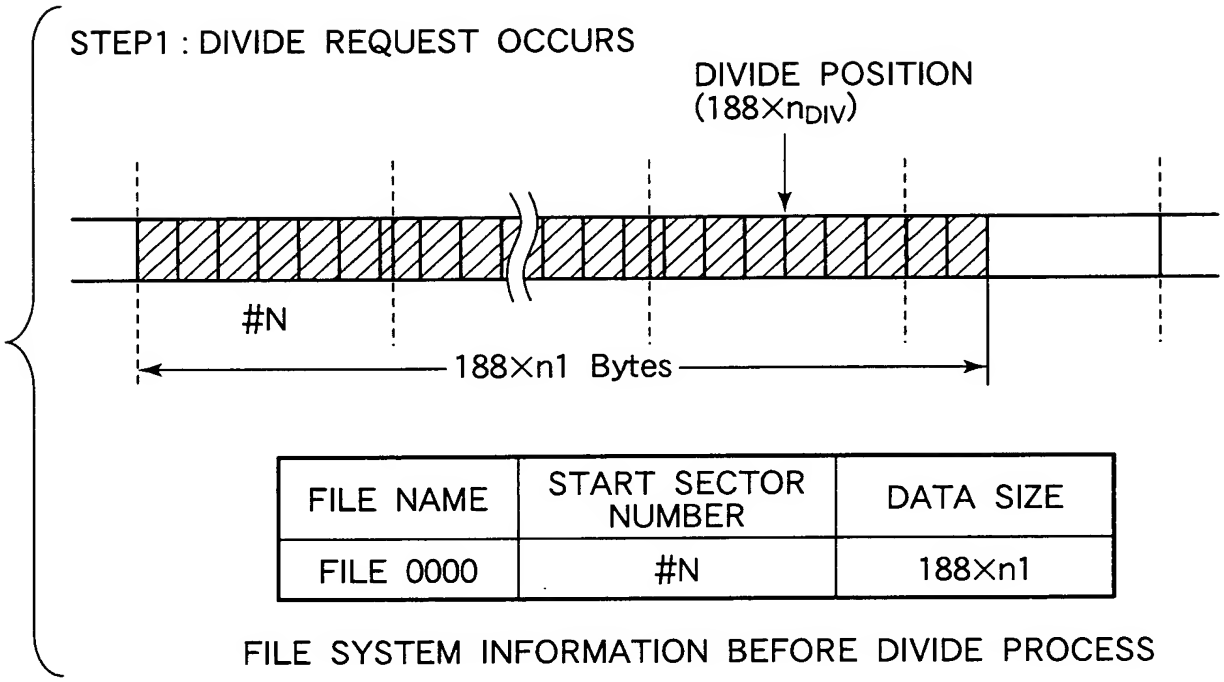


FIG.2

FILE NAME	START SECTOR NUMBER	ACCESS SIZE	FILE MAKING DATE
FILE A	A	$188 \times n_A$	yyyymmdd
FILE B	B1	$(188 \times n_B - X)$	yyyymmdd
	B2	X	...
FILE C	C1	$(188 \times n_C - Y)$	yyyymmdd
	C2	Y	...
...	...	...	...

FIG.3A

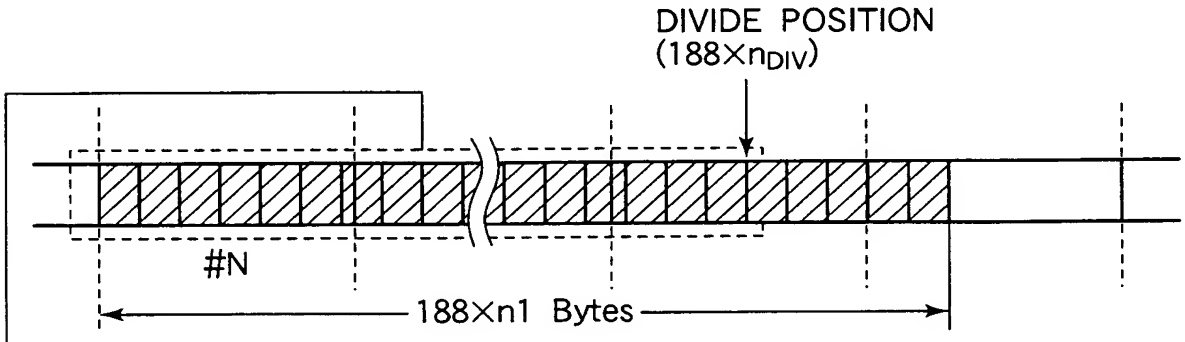


FILE NAME	START SECTOR NUMBER	DATA SIZE
FILE 0000	#N	188×n1

FILE SYSTEM INFORMATION BEFORE DIVIDE PROCESS

FIG.3B

STEP2 :
MAKE NEW FILE AS FILE 0001 EXTENDING UP TO DIVIDE POSITION



FILE NAME	START SECTOR NUMBER	DATA SIZE
FILE 0001	#N	188×nDIV

FIG.4A

STEP3 :
CALCULATE OFFSET L FROM DIVIDE POSITION TO BOUNDARY TO THE NEXT SECTOR
$L=2048-((188\times n_{DIV}) \bmod 2048)$

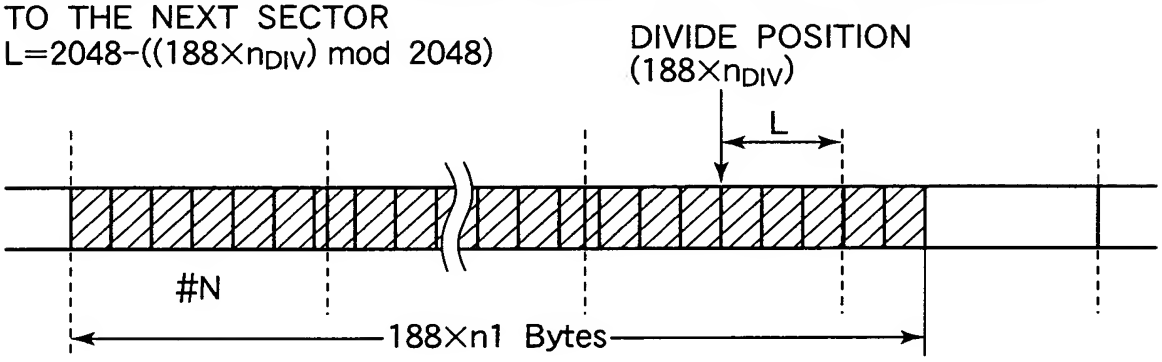


FIG.4B

STEP4 :
CALCULATE DUMMY TS PACKETS N_{NULL} FOR ADJUSTING ALIGNMENT BETWEEN TS PACKET BOUNDARY AND SECTOR BOUNDARY
WHEREIN N_{NULL} MEETS $(L+188\times n_{NULL}) \bmod 2048=0$

FIG.4C

STEP5 :
SEARCH NON-WRITTEN SECTOR NUMBER AND START SECTOR NUMBER #X, AND WRITE DUMMY TS PACKET

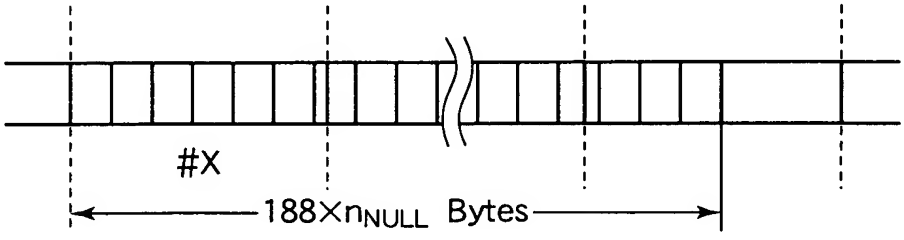


FIG.5A

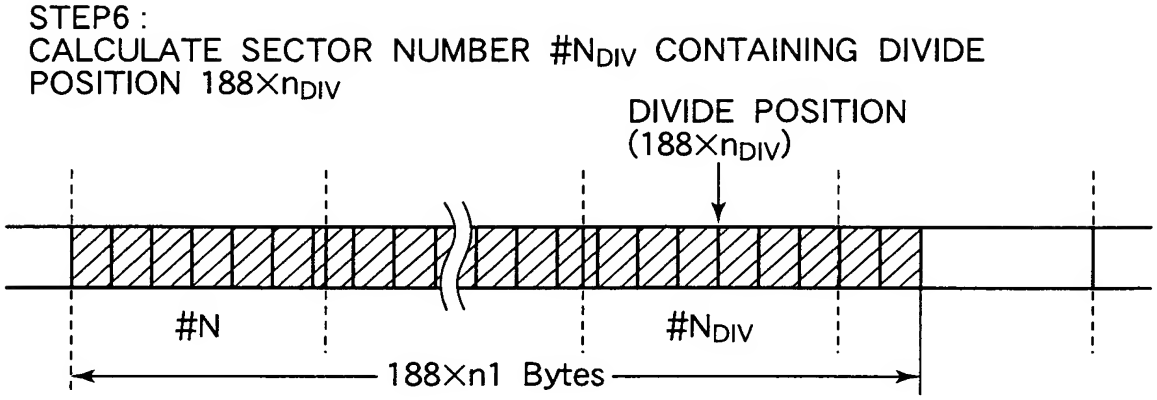

STEP6 :
CALCULATE SECTOR NUMBER #N$_{DIV}$ CONTAINING DIVIDE POSITION 188×n$_{DIV}$
DIVIDE POSITION
(188×n$_{DIV}$)
#N
#N$_{DIV}$
188×n1 Bytes

FIG.5B

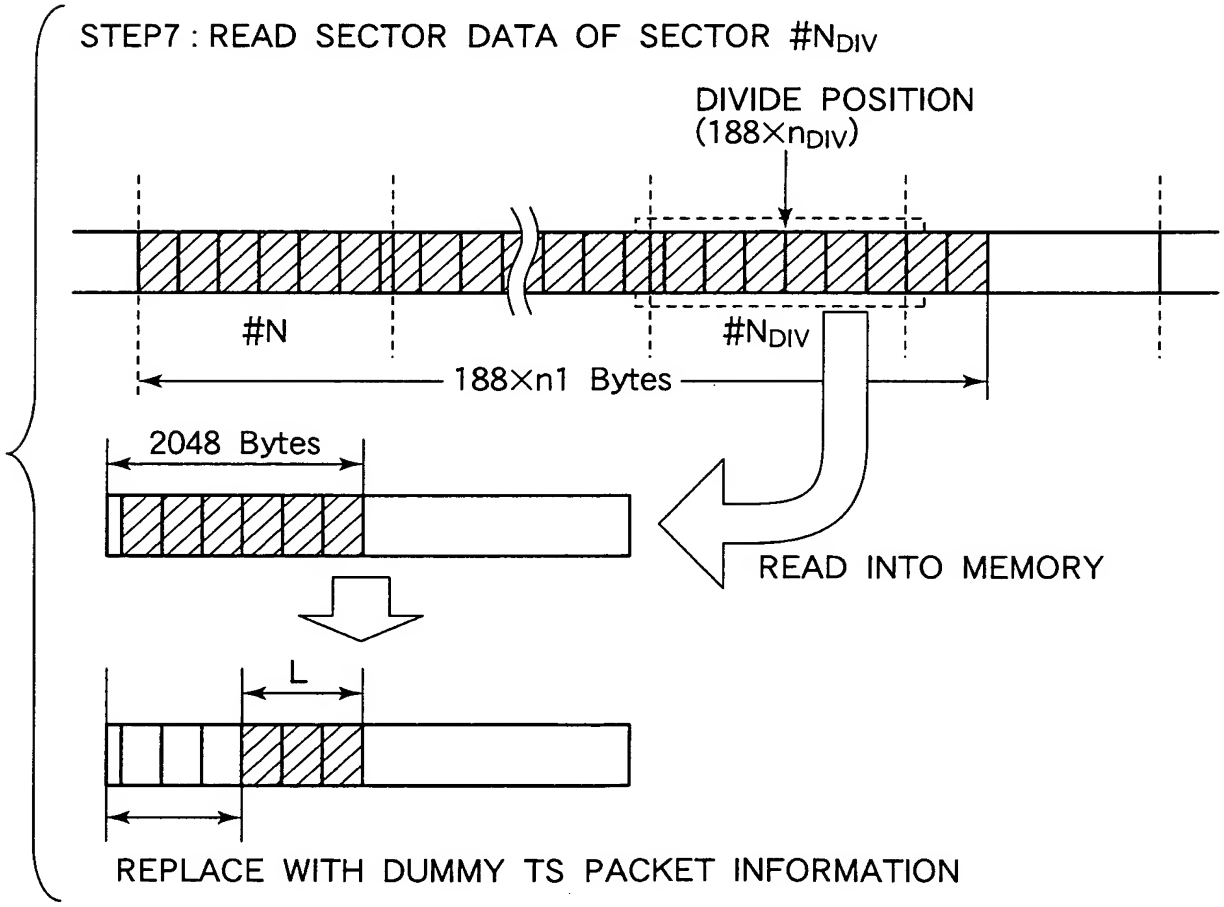
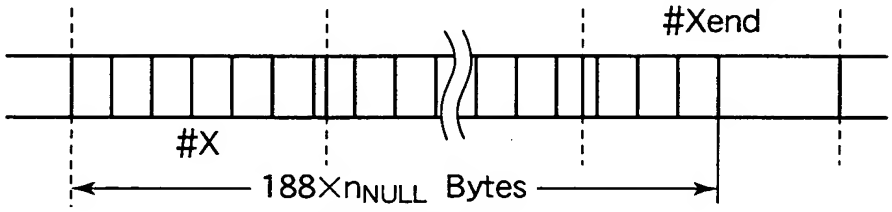

STEP7 : READ SECTOR DATA OF SECTOR #N$_{DIV}$
DIVIDE POSITION
(188×n$_{DIV}$)
#N
#N$_{DIV}$
188×n1 Bytes
2048 Bytes
READ INTO MEMORY
L
REPLACE WITH DUMMY TS PACKET INFORMATION

FIG.6A

STEP8 :
CALCULATE END SECTOR POSITION #Xend OF DUMMY TS PACKET



$\#Xend=\#X+(188\times n_{NULL})/2048$

FIG.6B

STEP9 :
OVERWRITE SECTOR DATA OF MEMORY ON SECTOR DATA #Xend

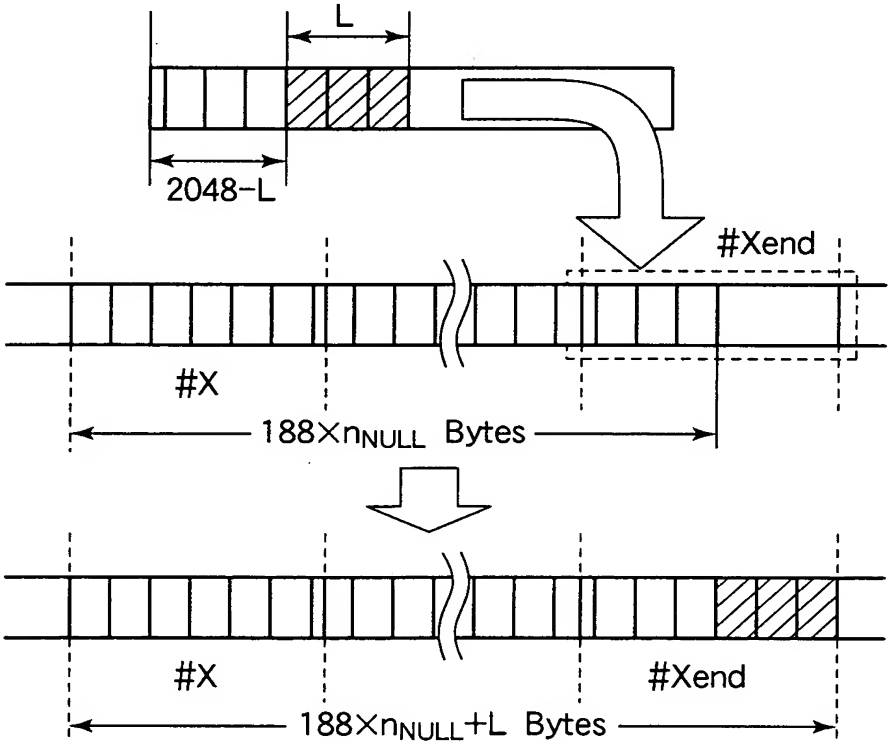


FIG.6C

STEP10 :
NEWLY ENTER DATA AS FILE 0002 FOLLOWING TO DIVIDE POSITION

FILE NAME	START SECTOR NUMBER	DATA SIZE
FILE 0001	#N	$188\times n_{DIV}$
FILE 0002	#X	$188\times n_{NULL}+L$
	$\#N_{DIV}+1$	$188\times n1-(188\times n_{DIV}-L)$

FIG.7

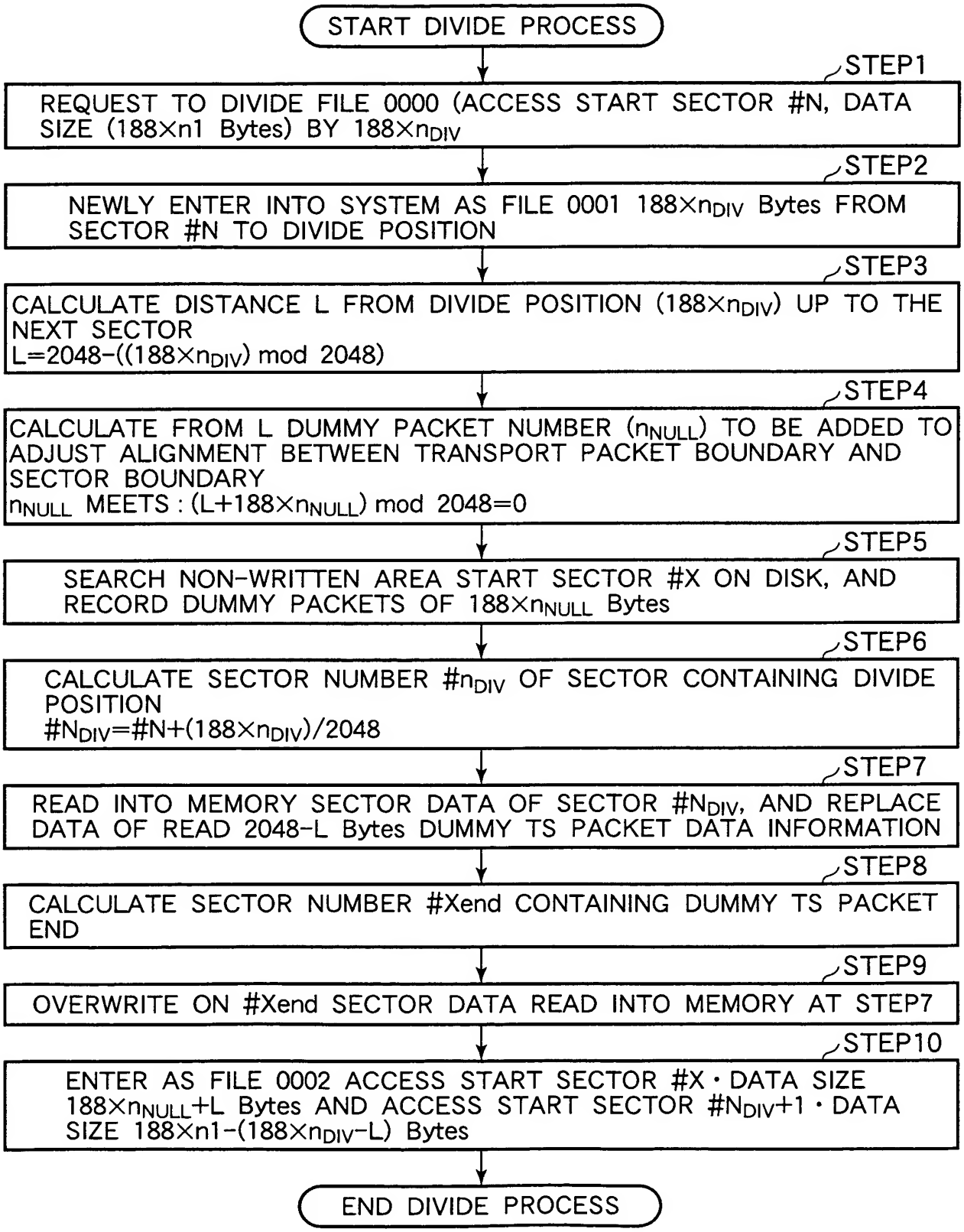

START DIVIDE PROCESS
STEP1
REQUEST TO DIVIDE FILE 0000 (ACCESS START SECTOR #N, DATA SIZE (188×n1 Bytes) BY 188×n_{DIV}
STEP2
NEWLY ENTER INTO SYSTEM AS FILE 0001 188×n_{DIV} Bytes FROM SECTOR #N TO DIVIDE POSITION
STEP3
CALCULATE DISTANCE L FROM DIVIDE POSITION (188×n_{DIV}) UP TO THE NEXT SECTOR
L=2048-((188×n_{DIV}) mod 2048)
STEP4
CALCULATE FROM L DUMMY PACKET NUMBER (n_{NULL}) TO BE ADDED TO ADJUST ALIGNMENT BETWEEN TRANSPORT PACKET BOUNDARY AND SECTOR BOUNDARY
n_{NULL} MEETS : (L+188×n_{NULL}) mod 2048=0
STEP5
SEARCH NON-WRITTEN AREA START SECTOR #X ON DISK, AND RECORD DUMMY PACKETS OF 188×n_{NULL} Bytes
STEP6
CALCULATE SECTOR NUMBER #n_{DIV} OF SECTOR CONTAINING DIVIDE POSITION
#N_{DIV}=#N+(188×n_{DIV})/2048
STEP7
READ INTO MEMORY SECTOR DATA OF SECTOR #N_{DIV}, AND REPLACE DATA OF READ 2048-L Bytes DUMMY TS PACKET DATA INFORMATION
STEP8
CALCULATE SECTOR NUMBER #Xend CONTAINING DUMMY TS PACKET
END
STEP9
OVERWRITE ON #Xend SECTOR DATA READ INTO MEMORY AT STEP7
STEP10
ENTER AS FILE 0002 ACCESS START SECTOR #X・DATA SIZE 188×n_{NULL}+L Bytes AND ACCESS START SECTOR #N_{DIV}+1・DATA SIZE 188×n1-(188×n_{DIV}-L) Bytes
END DIVIDE PROCESS

FIG.8

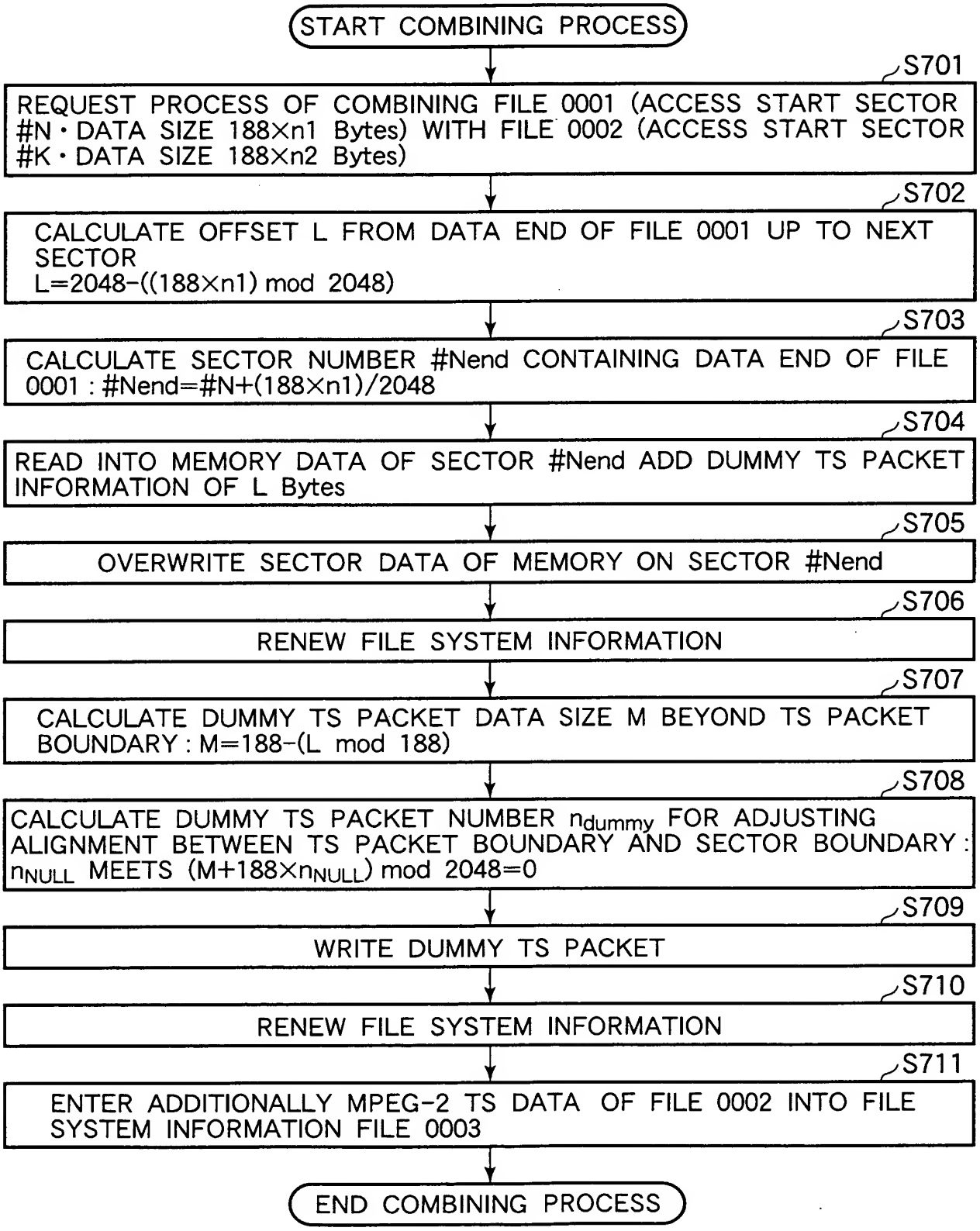
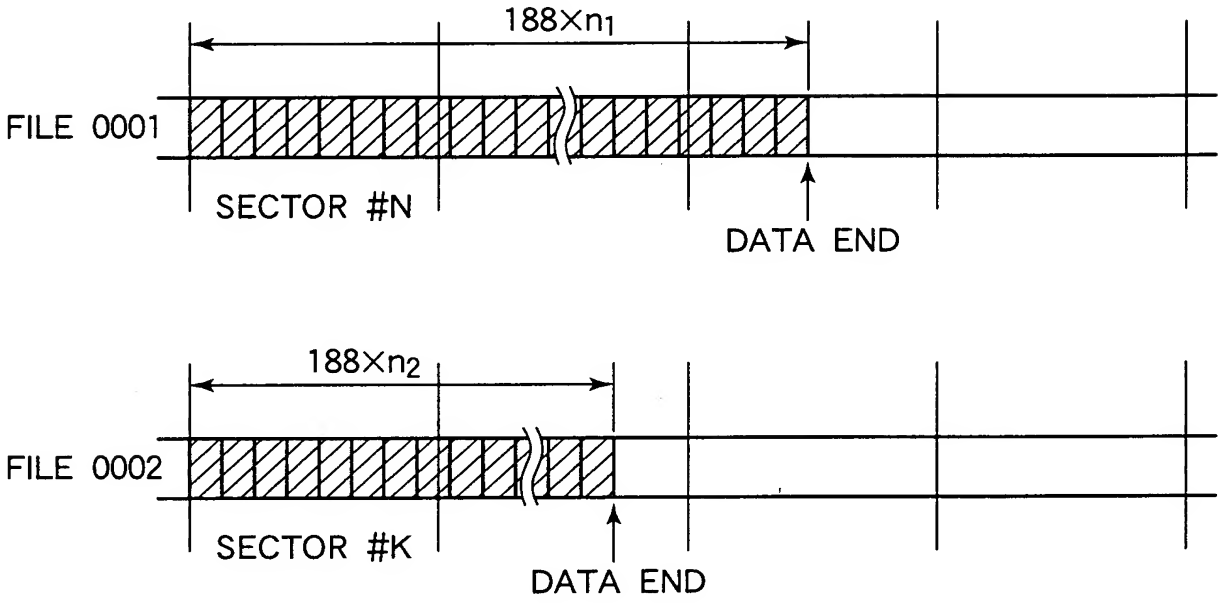

START COMBINING PROCESS
S701
REQUEST PROCESS OF COMBINING FILE 0001 (ACCESS START SECTOR #N・DATA SIZE 188×n1 Bytes) WITH FILE 0002 (ACCESS START SECTOR #K・DATA SIZE 188×n2 Bytes)
S702
CALCULATE OFFSET L FROM DATA END OF FILE 0001 UP TO NEXT SECTOR
L=2048-((188×n1) mod 2048)
S703
CALCULATE SECTOR NUMBER #Nend CONTAINING DATA END OF FILE 0001 : #Nend=#N+(188×n1)/2048
S704
READ INTO MEMORY DATA OF SECTOR #Nend ADD DUMMY TS PACKET INFORMATION OF L Bytes
S705
OVERWRITE SECTOR DATA OF MEMORY ON SECTOR #Nend
S706
RENEW FILE SYSTEM INFORMATION
S707
CALCULATE DUMMY TS PACKET DATA SIZE M BEYOND TS PACKET BOUNDARY : M=188-(L mod 188)
S708
CALCULATE DUMMY TS PACKET NUMBER n_{dummy} FOR ADJUSTING ALIGNMENT BETWEEN TS PACKET BOUNDARY AND SECTOR BOUNDARY : n_{NULL} MEETS (M+188×n_{NULL}) mod 2048=0
S709
WRITE DUMMY TS PACKET
S710
RENEW FILE SYSTEM INFORMATION
S711
ENTER ADDITIONALLY MPEG-2 TS DATA OF FILE 0002 INTO FILE SYSTEM INFORMATION FILE 0003
END COMBINING PROCESS

FIG.9

S701 : REQUEST COMBINING PROCESS



FILE NAME	START SECTOR NUMBER	DATA SIZE
FILE 0001	#N	188×n1
FILE 0002	#K	188×n2

FILE SYSTEM INFORMATION BEFORE COMBINING PROCESS

FIG.10A

S702 :
CALCULATE OFFSET L FROM DATA END OF FILE 0001 UP TO THE NEXT SECTOR

$$L=2048-((188\times n2) \bmod 2048)$$

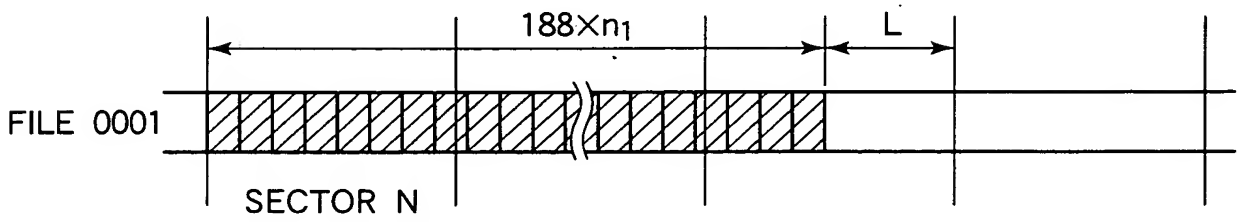


FIG.10B

S703 :
CALCULATE SECTOR NUMBER #Nend CONTAINING DATA END OF FILE 0001

$$\#Nend=\#N+(188\times n1)/2048$$

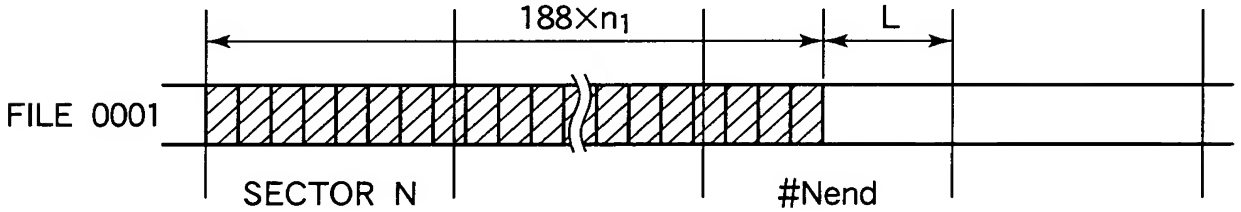


FIG.11

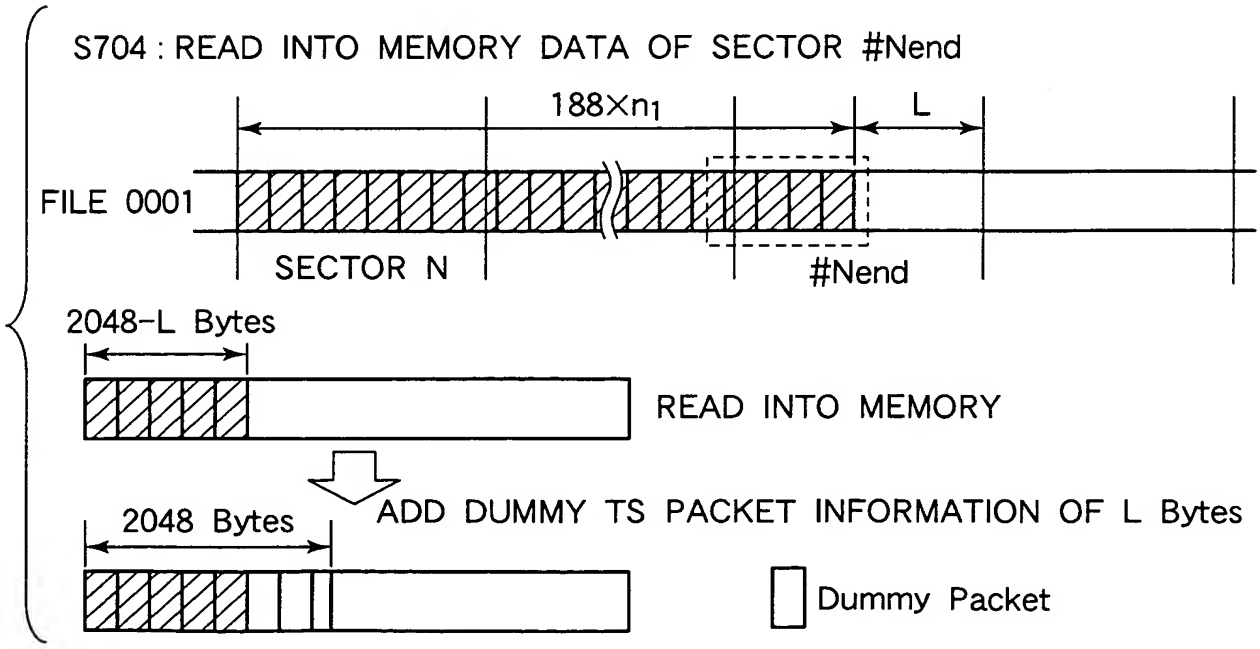


FIG.12A

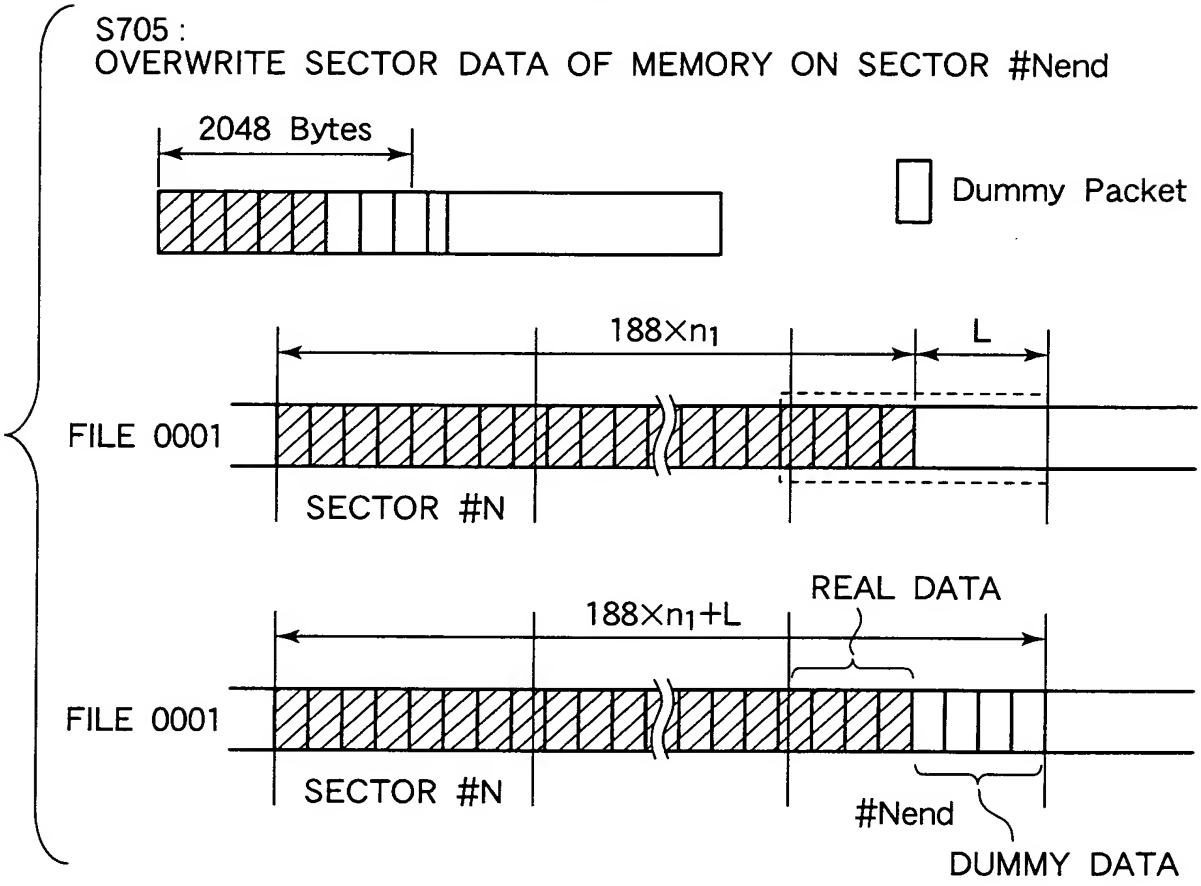


FIG.12B

S706 : RENEW FILE SYSTEM INFORMATION

FILE NAME	START SECTOR NUMBER	DATA SIZE
FILE 0003	#N	188×n1+L

FIG.12C

S707 :
CALCULATE DUMMY TS PACKET DATA SIZE M BEYOND TS PACKET BOUNDARY

$$M=188-(L \bmod 188)$$

FIG.13A

S708 :
CALCULATE DUMMY PACKET NUMBER N_{NULL} FOR ADJUSTING ALIGNMENT BETWEEN TS PACKET BOUNDARY AND SECTOR BOUNDARY

N_{NULL} MEETS : $(M+188\times N_{NULL}) \bmod 2048=0$

FIG.13B

S709 : READ DUMMY TS PACKET

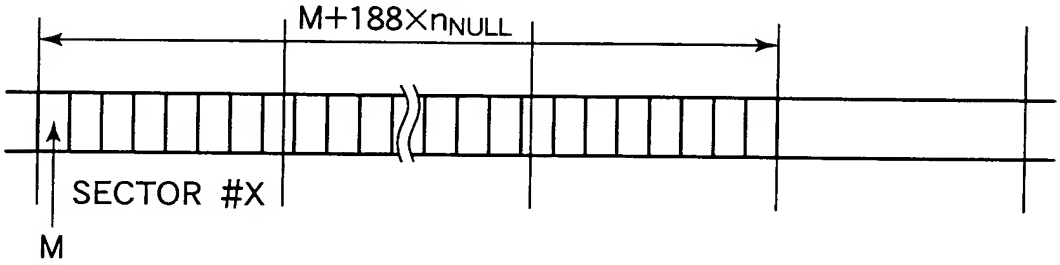
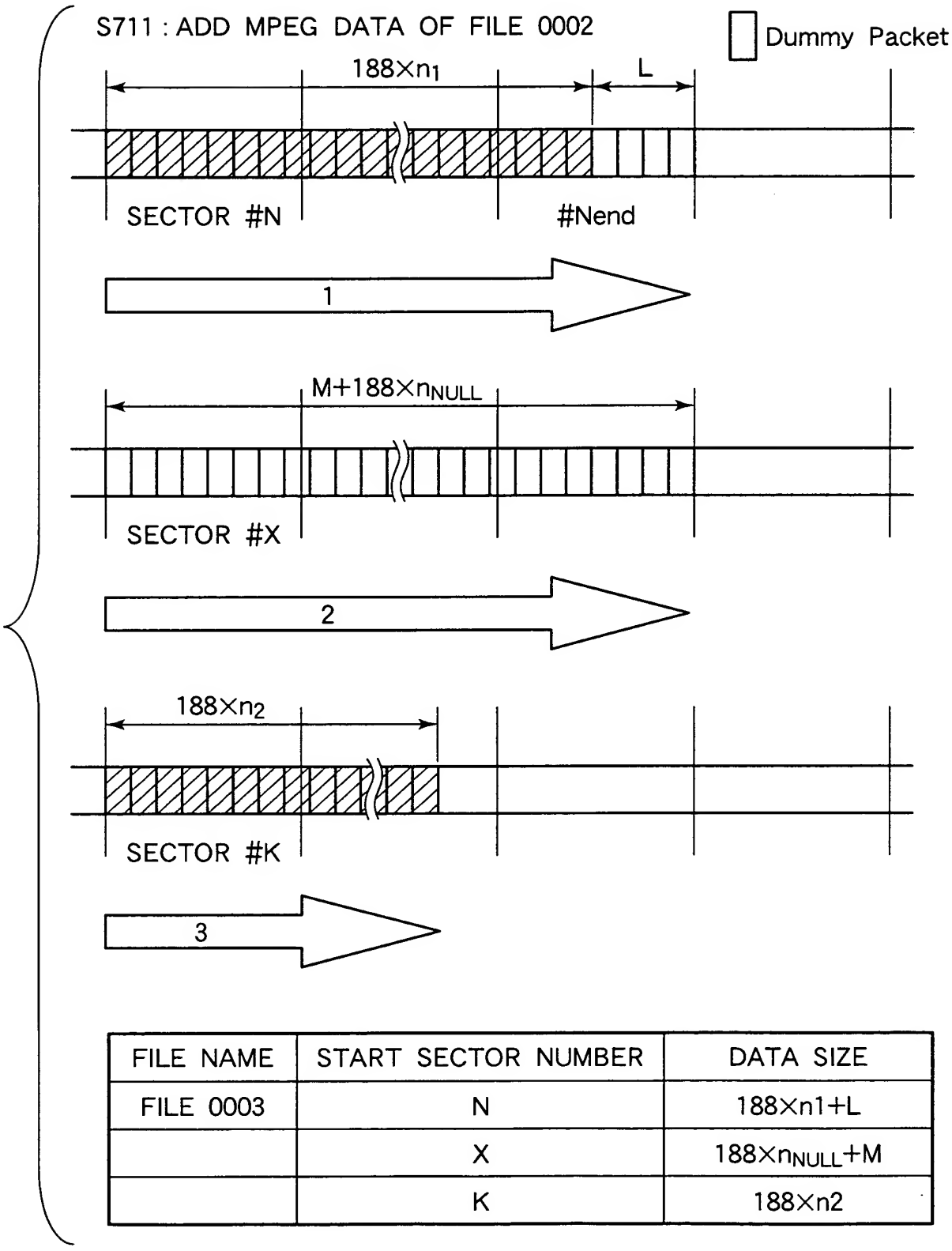


FIG.13C

S710 : RENEW FILE SYSTEM INFORMATION

FILE NAME	START SECTOR NUMBER	DATA SIZE
FILE 0003	#N	188×n1+L
	#X	$188\times n_{NULL}+M$

FIG.14



FILE NAME	START SECTOR NUMBER	DATA SIZE
FILE 0003	N	188×n1+L
	X	$188\times n_{NULL}+M$
	K	188×n2

FIG.15A

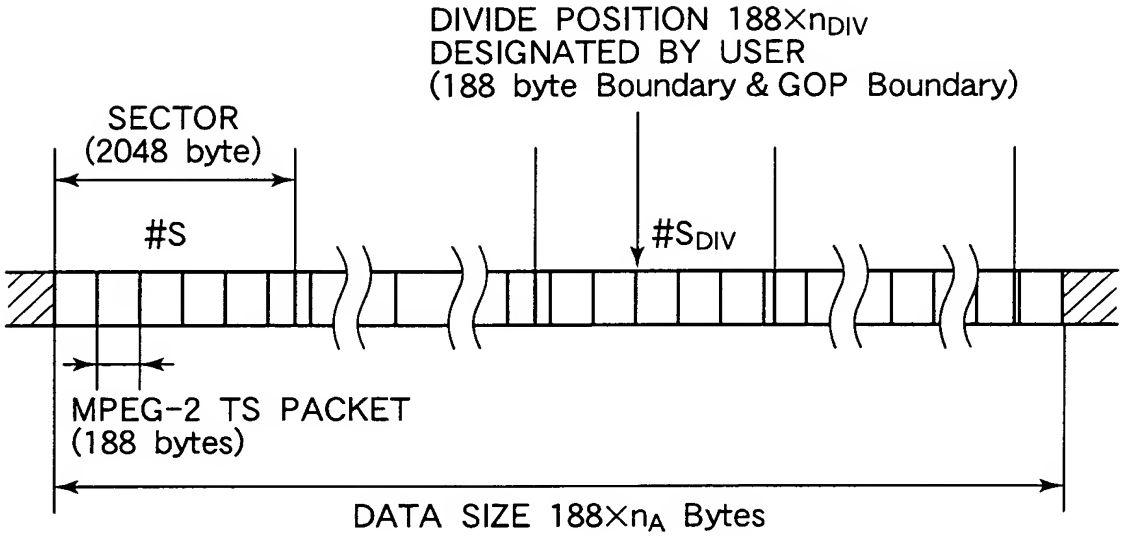


FIG.15B

FILE NAME	START SECTOR NUMBER	DATA SIZE
FILE_A	S	$188 \times n_A$

FIG.16A

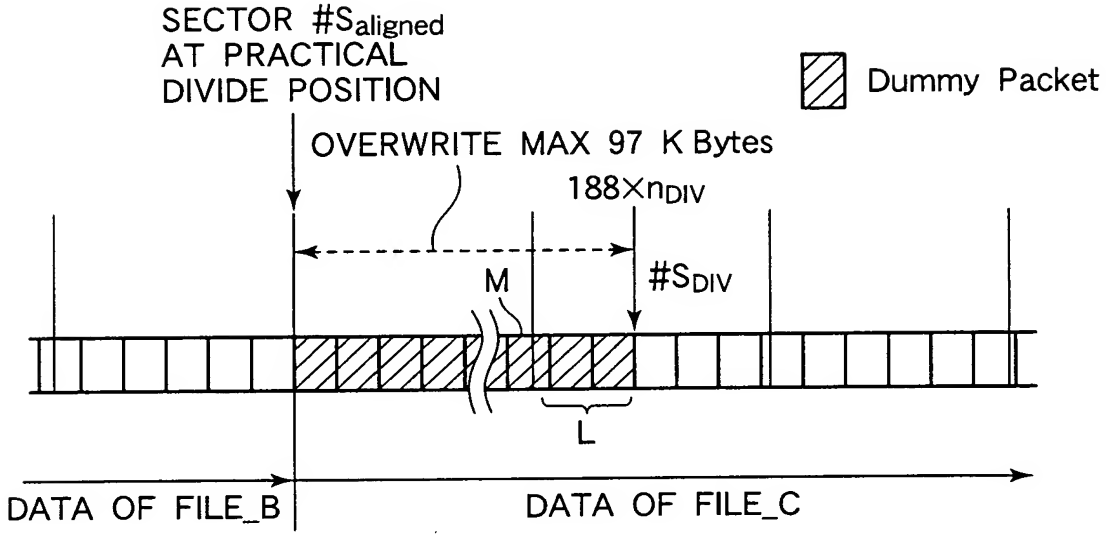


FIG.16B

FILE NAME	START SECTOR NUMBER	DATA SIZE
FILE_B	S	$(S_{aligned}-S)\times 2048$
FILE_C	$S_{aligned}$	$188\times n_A-(S_{aligned}-S)\times 2048$

FIG.17

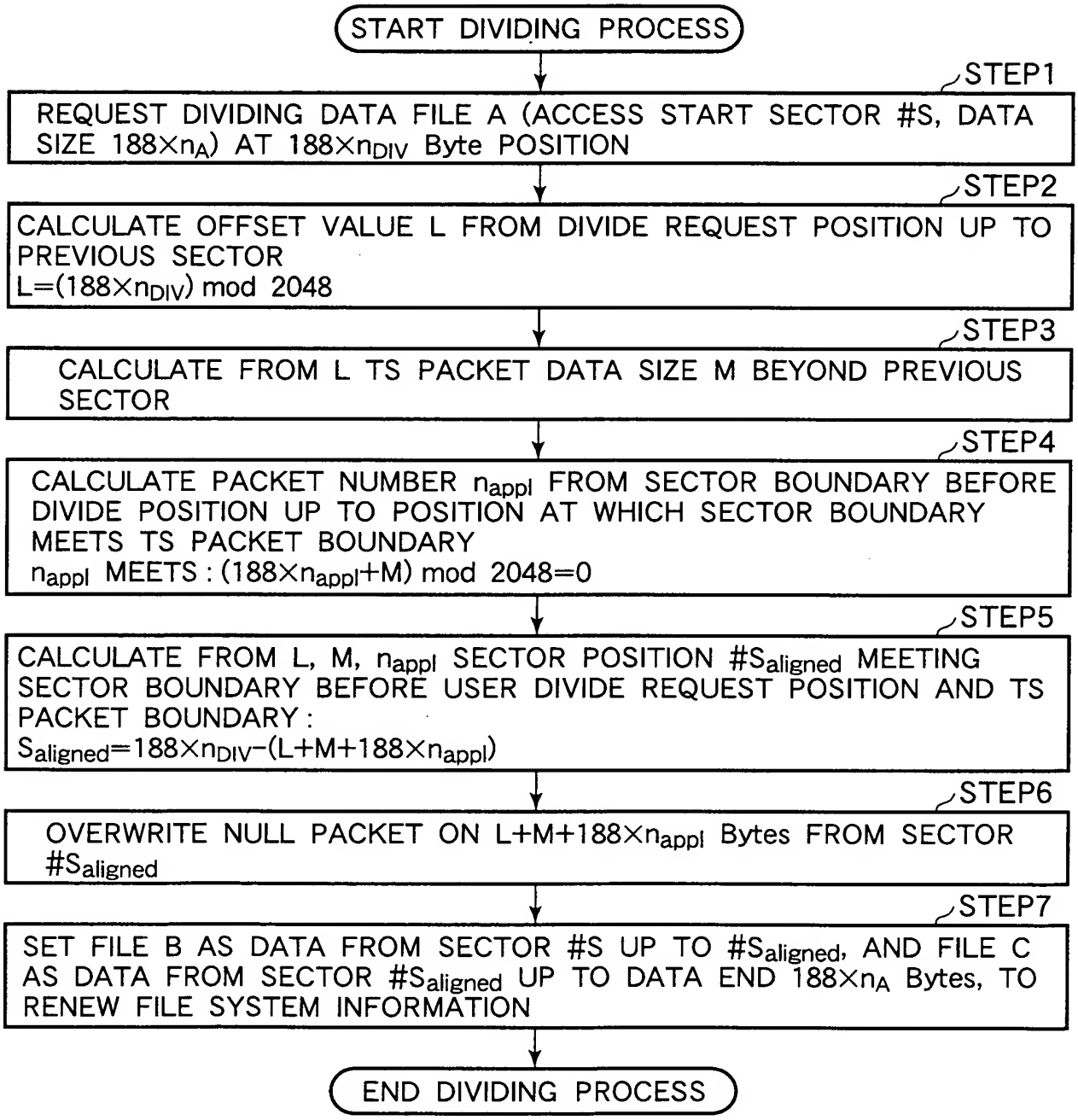

START DIVIDING PROCESS
STEP1
REQUEST DIVIDING DATA FILE A (ACCESS START SECTOR #S, DATA SIZE 188×n_A) AT 188×n_{DIV} Byte POSITION
STEP2
CALCULATE OFFSET VALUE L FROM DIVIDE REQUEST POSITION UP TO PREVIOUS SECTOR
L=(188×n_{DIV}) mod 2048
STEP3
CALCULATE FROM L TS PACKET DATA SIZE M BEYOND PREVIOUS SECTOR
STEP4
CALCULATE PACKET NUMBER n_{appl} FROM SECTOR BOUNDARY BEFORE DIVIDE POSITION UP TO POSITION AT WHICH SECTOR BOUNDARY MEETS TS PACKET BOUNDARY
n_{appl} MEETS : (188×n_{appl}+M) mod 2048=0
STEP5
CALCULATE FROM L, M, n_{appl} SECTOR POSITION #$S_{aligned}$ MEETING SECTOR BOUNDARY BEFORE USER DIVIDE REQUEST POSITION AND TS PACKET BOUNDARY :
$S_{aligned}$=188×n_{DIV}-(L+M+188×n_{appl})
STEP6
OVERWRITE NULL PACKET ON L+M+188×n_{appl} Bytes FROM SECTOR #$S_{aligned}$
STEP7
SET FILE B AS DATA FROM SECTOR #S UP TO #$S_{aligned}$, AND FILE C AS DATA FROM SECTOR #$S_{aligned}$ UP TO DATA END 188×n_A Bytes, TO RENEW FILE SYSTEM INFORMATION
END DIVIDING PROCESS

FIG.18

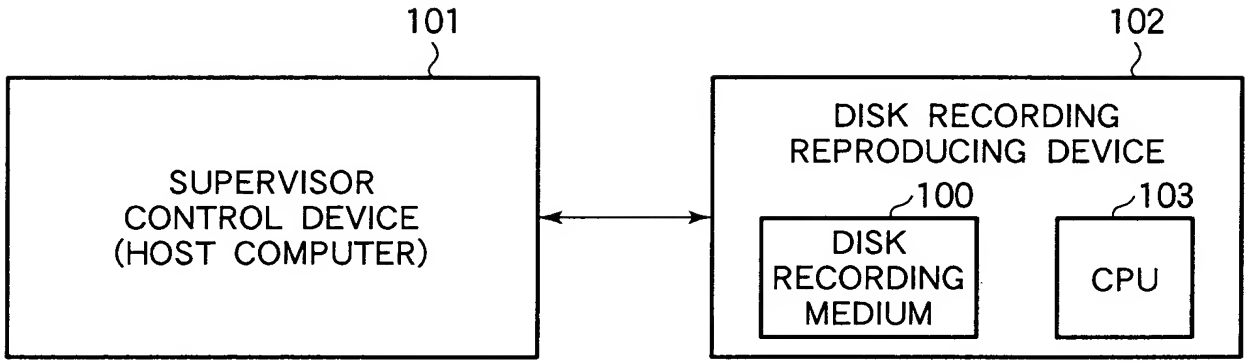

101
SUPERVISOR
CONTROL DEVICE
(HOST COMPUTER)
102
DISK RECORDING
REPRODUCING DEVICE
100
DISK
RECORDING
MEDIUM
103
CPU

FIG.19

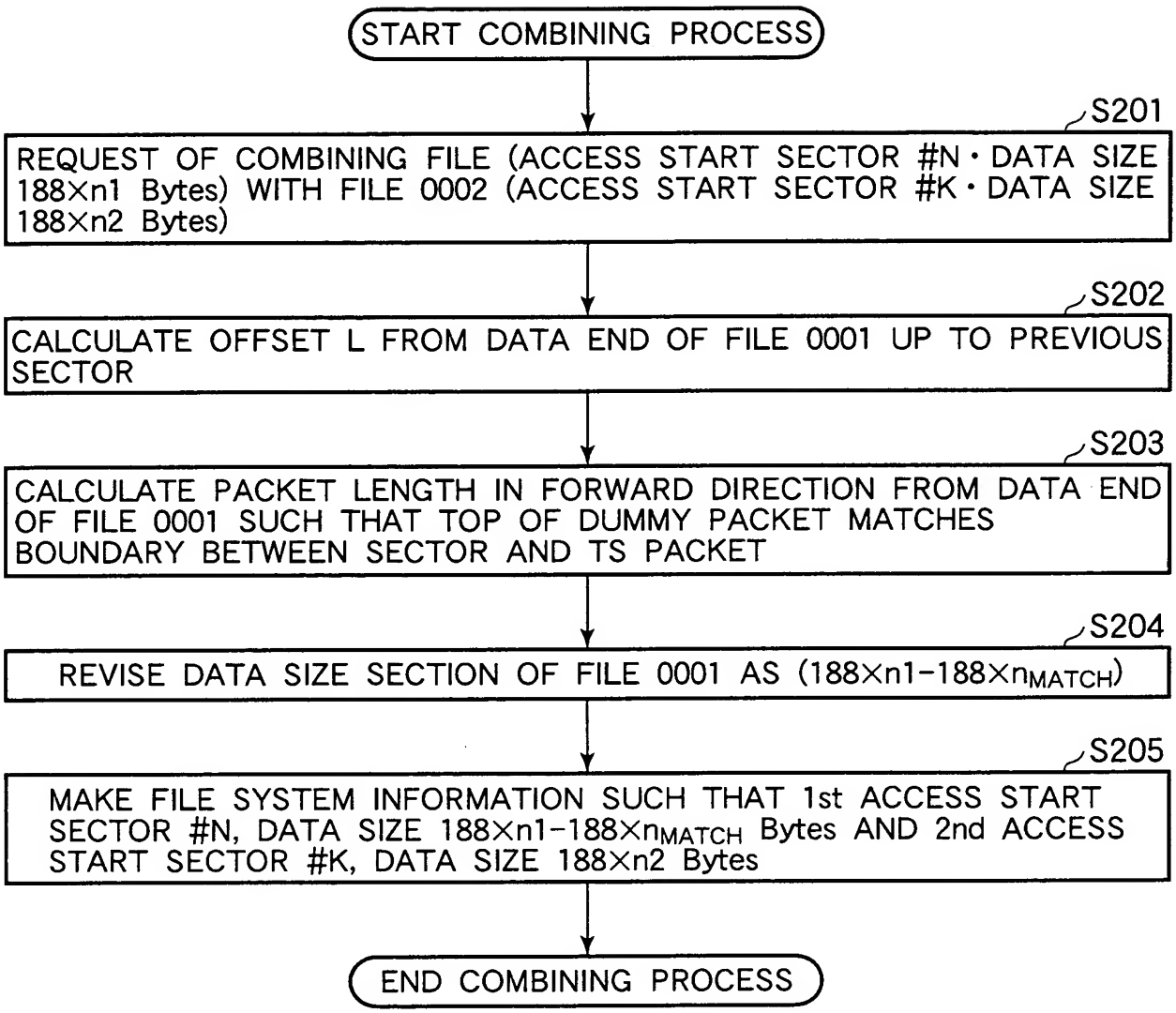
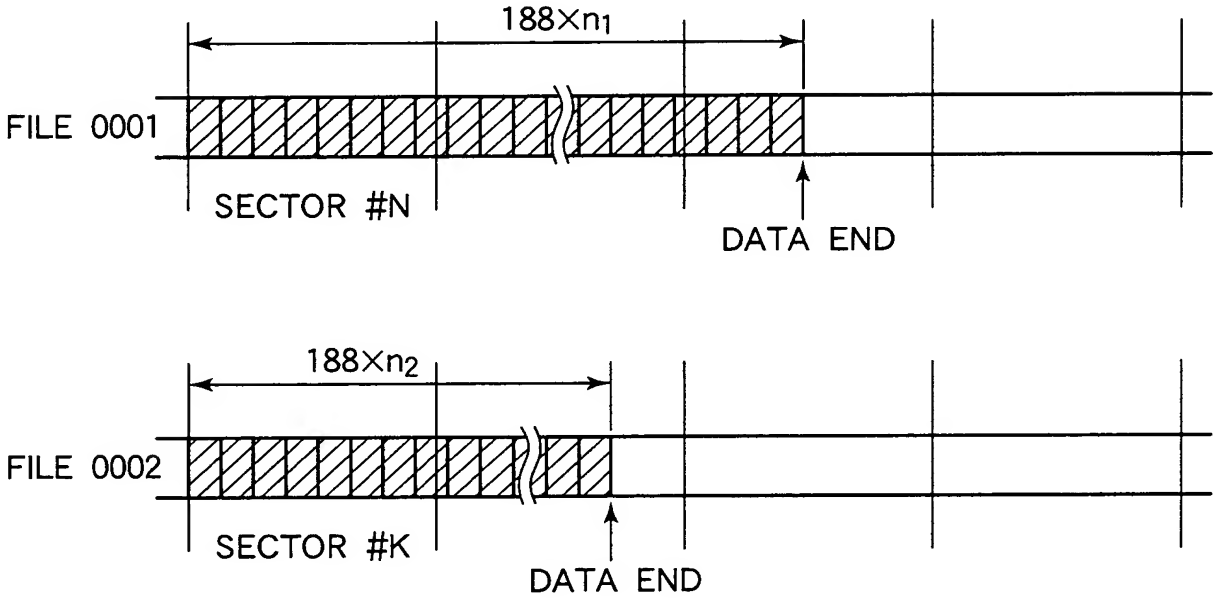

START COMBINING PROCESS
S201
REQUEST OF COMBINING FILE (ACCESS START SECTOR #N・DATA SIZE 188×n1 Bytes) WITH FILE 0002 (ACCESS START SECTOR #K・DATA SIZE 188×n2 Bytes)
S202
CALCULATE OFFSET L FROM DATA END OF FILE 0001 UP TO PREVIOUS SECTOR
S203
CALCULATE PACKET LENGTH IN FORWARD DIRECTION FROM DATA END OF FILE 0001 SUCH THAT TOP OF DUMMY PACKET MATCHES BOUNDARY BETWEEN SECTOR AND TS PACKET
S204
REVISE DATA SIZE SECTION OF FILE 0001 AS (188×n1−188×nMATCH)
S205
MAKE FILE SYSTEM INFORMATION SUCH THAT 1st ACCESS START SECTOR #N, DATA SIZE 188×n1−188×nMATCH Bytes AND 2nd ACCESS START SECTOR #K, DATA SIZE 188×n2 Bytes
END COMBINING PROCESS

FIG.20

S201 : COMBINING PROCESS REQUEST OCCURS



FILE NAME	START SECTOR NUMBER	DATA SIZE
FILE 0001	#N	188×n1
FILE 0002	#K	188×n2

FILE SYSTEM INFORMATION BEFORE COMBINING PROCESS

FIG.21A

S202 :
CALCULATE OFFSET L FROM DATA END OF FILE 0001 UP TO PREVIOUS SECTOR

$$L=(188\times n1) \bmod 2048$$

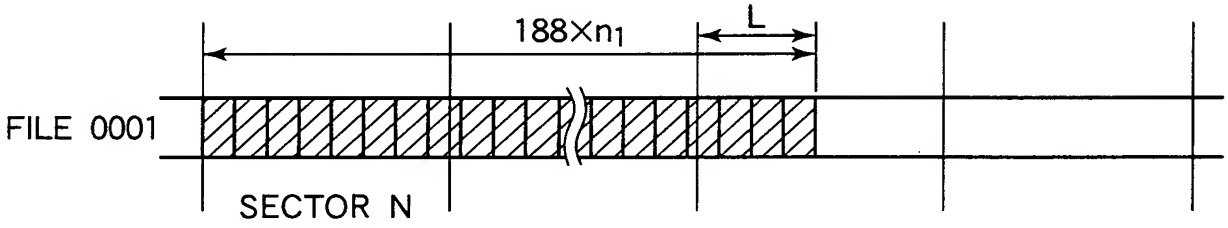


FIG.21B

S203 :
CALCULATE LENGTH IN FORWARD DIRECTION FROM DATA END OF FILE 0001 TO COMMON BOUNDARY SUCH THAT PACKET BOUNDARY MATCHES SECTOR BOUNDARY :

n_{MATCH} MEETS $((188\times n_{MATCH})-L) \bmod 2048=0$

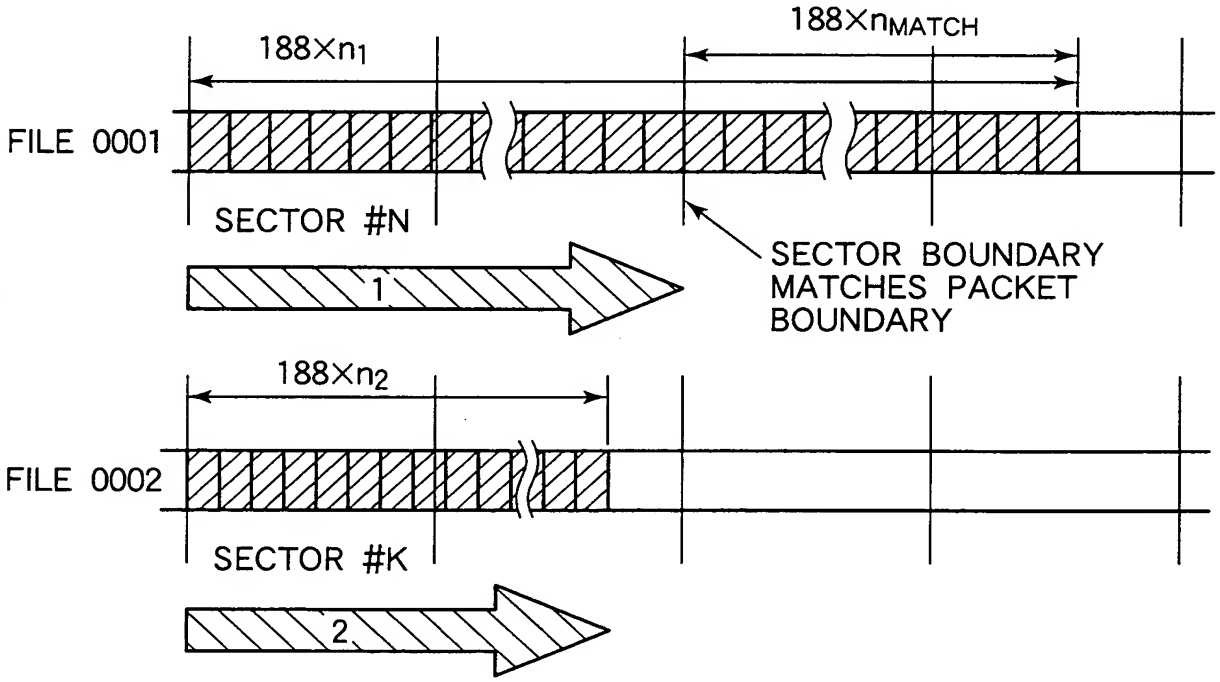
FIG.22A

S204 :
REVISE DATA SIZE SECTION OF FILE 0001 IN FILE SYSTEM INFORMATION

FILE NAME	START SECTOR NUMBER	DATA SIZE
FILE 0001	#N	$188\times n1-188\times n_{MATCH}$
FILE 0002	#K	$188\times n2$

FIG.22B

S205 : RENEW FILE SYSTEM INFORMATION FOR COMBINING



FILE NAME	START SECTOR NUMBER	DATA SIZE
FILE 0003	1st ACCESS START SECTOR NUMBER #N	$188\times n1-188\times n_{MATCH}$
	2nd ACCESS START SECTOR NUMBER #K	$188\times n2$

DATA LENGTH OF FILE 0001 BEFORE COMBINING CAN BE DIVIDED BY 2048 Bytes :

$$(188\times n1-188\times n_{MATCH}) \bmod 2048=0$$